Title: System and Method for Multicast Stream Failover

Inventors: Joo C. Chung, Michael Sun, Kenneth Gould and Frank Huang

Field of the Invention:

1. This invention relates generally to broadcasting streaming data over a network. More particularly, the present invention is a system for avoiding a single point failure when multicasting a data stream comprising video, audio and other data.

Background of the Invention:

2. Multicast streaming services provide for the continual feed of streaming data, including video, over a network to many different users. The most common application is streaming video across a network system. Audio streaming is another common application. However, these streaming application types are not intended as a limitation. Typically, a particular data stream is multicast from a single server over a network. This allows many people to receive and view the same program simultaneously.

3. If however, the server becomes unavailable, then the multicast stream which is being served from the server will also be unavailable, thus leading to an outage of service.

4. Increasingly, the lines between cable television service and serving a data stream over the Internet is becoming blurred. More and more, television-like content is being served to users of the Internet. This is becoming increasingly so in an era where bandwidth is increasing, and is readily available to Internet users. Today, Internet cable, DSL, and other types of high bandwidth lines are readily available to Internet users. Such users are demanding more and more content.

5. When a web based asset is used to stream video to Internet users, especially multiple users, there are generally two schemes that are employed. First, a unicast of the data stream is made to a single user. In a unicast situation, a single user connects to a website having program content. Upon demand, that program content is served by the web server to the individual user. When multiple users desire a particular program at different times, multiple unicasts of the data are made by the server. However, a single server can only serve a finite number of unicast streams of a given program content.

6. Multiple streaming servers are used to load balance unicast data to many individual users. If a particular server fails, each user suffers an outage, albeit of differing content initiated at different times. Frequently multiple streaming servers are used in order to achieve load balancing and to allow multiple unicast streams of the same web asset. This approach works well when an equal number of video streams are served from each of the servers. However, if any server goes down, the individuals who are being served from that server will have their service disrupted.

7. When a streaming server is distributing a video stream in a multicast mode (that is, simultaneous streaming of a video stream by multiple users), rather than in multiple unicast streams, the load on the server is relatively low. This is because a single stream per program is being broadcast by the server to multiple users, therefore multiple interactions with the server are not required for a single program. In a multicast situation, load balancing is not as much of an issue and therefore, multiple servers are not required. These relatively unused stream server resources can be used to increase the program availability by providing redundant servers of the same content

8. What would be particularly useful is a system which allows for the unicast or multicast of a video stream with the reliability that is associated with redundancy in the system.

Summary of the Invention:

9. The present invention is a system and method for harnessing the relatively under utilized streaming servers as redundant servers to minimize user outage due to a server failure particularly under multicast mode.

10. It is therefore an objective of the present invention to improve the reliability of multicast streaming service over a video cable network.

11. It is yet another objective of the present invention to improve the reliability of multicast streaming service over networks serving clients.

12. It is a further objective of the present invention to improve the reliability of unicast broadcasts over client networks and over video networks.

13. It is a further objective of the present invention to introduce redundancy into the network serving multicast customers to improve reliability.

14. It is yet another objective of the present invention to ensure that service is not interrupted to cable and network clients who are receiving stream data applications.

15. The present invention is a system and method that is used in conjunction with redundant multicast servers to enhance the reliability of receipt of data stream desired. The use of two servers is the preferred embodiment although this is not meant as a limitation. More than two servers can be used although it is anticipated that there will not be a corresponding increase in reliability over the architecture that uses two multicast servers. One server broadcasts the primary stream. The other server broadcasts the

secondary stream. The secondary stream data content is identical with the primary stream data content.

16. Each server multicasts the same content at the same time over the same enterprise network that is connected to the monitoring device of the present invention. An enterprise network may be a local area network, wide area network, intranet or any other network whereby the enterprise telecommunicates internally. For purposes of this application, the present invention is a system and method which includes a device that monitors for an adverse change of the primary multicasting server to utilize the secondary stream content when an adverse change is detected. An adverse change is a detected error which may range from corruption in the packet to the absence of the packet itself. This device multicasts the secondary stream content to the users when the primary stream server undergoes an adverse change, including failure. Hence, the device that monitors the two servers is termed a "failover" device.

17. As noted above, each server multicasts the same content at the same time over the same enterprise network connected to the failover device. The multicast stream from each server is assigned a different IP address and port number. The failover device is configured to select from one of the multicast streams and designates that stream as the primary stream. The failover device designates the second multicast stream as the secondary stream.

18. The failover device listens for both multicast streams. The device buffers the primary multicast stream packets as well as the secondary multicast stream packets. Under normal conditions, that is no adverse change is detected, the failover device takes a copy of the primary stream data packet to multicast. In the event of a primary stream

adverse change, the failover device takes a copy of secondary multicast server to multicast. The failover device overwrites the IP address and Port number in the packet it will multicast with its own virtual IP address and virtual port number. This virtual IP header information replaces the real IP header information that was in the original data packet. The rewritten data packet is then multicast by the failover device. The program data content (other than the IP header information) multicast by the failover device is therefore unaltered from the source data packet. As noted above, the secondary multicast stream packets are buffered but not otherwise used until an adverse change from the primary stream packet is detected.

19. In this fashion, there is a minimal loss of content to the clients whether the adverse changes are continuous or occasional.

20. The system of the present invention can be used by any application that employs multicast streaming, including, but not limited to video and audio programming. Further, the present invention finds its best use for multicast traffic which does not have stringent synchronization requirements and which can tolerate a minimal loss of data. Video and audio streams are examples of the type of content that is well suited for this failover methodology.

21. It should also be noted that while the failover device buffers and multicasts the packets, the failover device itself does not synchronize the multicast stream that is being broadcast by the multiple servers of the present invention. This is separately synchronized between the multicast servers. It does, however, synchronize the packets of data that are to be subsequently multicast.

22. The failover device may use the packet count information present in the packet header to synchronize the packets received from the primary and secondary multicast servers. This synchronization method is one embodiment and is not meant to be limiting.

Brief Description of the Drawings:

23. Figure 1 illustrates the functioning of the failover device.

24. Figure 2 illustrates the buffering and switching associated with the failover device.

Detailed Description of the Invention:

25. As noted above, the present invention comprises a failover device monitoring redundant streaming servers in order to ensure reliability and uninterrupted service for viewers of video programs.

26. Referring first to **Figure 1**, the functioning of the failover device is illustrated. Multicast server A **10** and multicast server B **12**, each multicast the same content over the same enterprise network **14**. The two multicast servers each constitute a Real Multicast transmitting a multicast IP address and Port (RMIPP). Each multicast server multicasts stream packets having IP headers associated with the address and port of the associated server. Each RMIPP is unique in order to avoid duplicate network traffic. Thus, in **Figure 1**, RMIPP-A represents the multicast channel from multicast server A over which a particular data stream is broadcast. The same video stream is multicast over RMIPP-B which represents the channel over which the same data stream is broadcast but, in this case, from multicast server B **12**.

27. The multicast content is received at the multicast stream failover device **16** which comprises logic to select one of the multicast streams as the primary stream and the other stream as the failover or secondary stream.

28. The failover device buffers the primary multicast stream packets in buffer **18**. Packets received from the secondary stream are buffered in buffer **20** (as more fully explained below). The failover device **16** can distinguish the source of the packets because the IP header information in each packet contains unique RMIPP. If there are no adverse changes detected in the primary data packet, the failover device overwrites the IP header information with its own Virtual Multicast IP address and Port number (VMIPP) in the primary stream packet. If an adverse change is detected, the failover device selects the secondary stream packet, overwriting the IP header information with its own VMIPP.

29. Thereafter the failover device multicasts the packet comprised of rewritten packets with the new virtual multicast IP address and port number. The source of the packet content is from the primary multicast server if there are no detectable adverse changes in the received packet. Otherwise the source is the secondary multicast server. The failover device will synchronize the packets from the two multicast servers such that the next packet multicast, regardless of the content source, is the packet that sequentially follows the last packet multicast. Thus, the multicast failover device forwards the content stream from either RMIPP-A or RMIPP-B where the IP header is rewritten as VMIPP.

30. Synchronization of the two video streams is carried out by the servers in communication with one another. Thus, the failover device performs a failover from one RMIPP to another RMIPP in the event that the primary RMIPP has evidence of some adverse change. As noted above, adverse change includes the loss of a packet from a

multicast stream as well as corruption of the data. Two multicast sources are synchronized in time and have access to the same source data and programming instructions (either via a shared disk **8** or from independent replicas of the data). By virtue of this synchronization, the servers **10, 12** multicast the same data (differing only in data source information) at the same time. Allowing both sources to be exact replicas makes configuration much simpler and does not require any additional development work to integrate into an architecture supporting Windows Media video multicasts.

31. Multicast sources **10, 12** can be connected via separate physical ports or the same port depending on what mechanism is used to perform the actual filtering (L2/L3 sourceID filters vs physical port filters).

32. To provide accurate filtering logic, the system of the present invention optionally provides a Monitor **21** with access to the Programming data from the shared disk **8** (which is preferably in a reduced form which comprises the address and expected bitrate of every active multicast in a given time window).

33. An optional "sniffer" hub **19** provides a higher level of certainty to the monitor via a second NIC card. Capturing duplicates of all backnet **14** traffic provides the Monitor **21** with a guaranteed view of the backnet **14** status. The certainty comes from additional knowledge concerning whether data loss is truly occurring at the multicast sources versus at the net, switch, or NIC card. It also allows the Monitor **21** to know if switching between sources will actually help (i.e. it can also monitor the non-active (redundant-mode) source whose packets are not forwarded to the frontnet **15** for subsequent distribution to receivers **26, 28, 30, 32,** and **34**).

34. Multicast packets on a particular address are forwarded from only one source at a time (i.e. redundant packets from secondary source are filtered out -- there is no packet rewriting). Note that while not necessary as long as receiving clients use buffers and can recognize duplicate and misordered packets, it is possible to provide packet-level synchronization with buffering and perhaps some knowledge of the packet format.

35. The Monitor **21** can control the failover mechanisms in the failover device **16** switch via exposed SNMP controls **24**. Threshholding logic in the failover device **16** dictates when such a failover should occur and failover may be at the individual multicast stream or entire multicast source level. Note that while an external Monitor **21** is illustrated, this is not meant as a limitation. For example, the same functionality of filtering duplicate packets based on L2/L3 sourceID or physical port can be implemented within the failover device **16** which can be more flexible since it does not rely upon the limited set of SNMP controls.

36. Windows Media player, whose capabilities are incorporated herein by reference in their entirety, is capable of handling duplicate and misordered packets via a buffering mechanism so as to overcome the lack of sync at the packet level that would become apparent when a stream is rolled over between sources.

37. Referring to **Figure 2**, the functioning and buffering of multicast content is illustrated. Primary streaming server **10** streams its data content using IP address X.X.X.X at port XX. Secondary streaming server **12** multicasts its data content, which is the same content and is synchronized with the content of primary streaming server **10**, using IP address Y.Y.Y.Y at port YY. As noted earlier, these video streams are

synchronized with one another. Each video stream, from the primary streaming server and the secondary streaming server, is transmitted over the same enterprise network **14** to the failover device **16**.

38. The work that is sent to the customer, whether audio, video or other work, is the collection of data packets played in a strict sequence. While the packets may be delivered out of sequence, the playing device buffers the packets and orders the playing of the packets according to the packet sequence number found in the packet header area. For purposes of this description, packet content refers to the data viewed or heard by the end user, whereas header data includes IP addresses, port numbers and packet sequence numbers. As discussed below, packet sequence numbers are used to order the packets within the failover device buffers.

39. Multicast failover device **16** buffers the packet stream both from the primary streaming server **10** and from the secondary streaming server **12**. The packets from each are buffered such that the multicast failover device at a point in time has packet with sequence number X from the primary multicast stream and a packet from the secondary multicast stream with identical content and the same sequence number X in a second buffer. Further, the failover device **16** also has packet X+1, packet X+2, packet X+3, packet X+4, etc. from the primary multicast stream as well as packet X+1, packet X+2, packet X+3, and packet X+4, etc. from the secondary multicast stream.

40. While the multicast failover device does not synchronize the output of the primary and secondary streaming servers, **10, 12**, it does synchronize the packets received so that the packets in the failover device buffers, at any point in time, contain the

same data content packets in each buffer in the same order. Each buffer's content, assuming no packet defects, should be identical in the respective buffers at the same time.

41. One method for packet synchronization at the failover device is to use this packet sequence number contained in the packet's header. Each transmitting server numbers each packet. The failover device inserts the received packets into the buffer at a buffer index location corresponding to the packet's number. For example, packet number X from the primary server will be placed in the buffer for the primary server at index location equal to X.

42. Since programs are sufficiently large, the buffers are periodically recycled and overwritten. The packet number would be mapped by logic that converts the sequence number to an index value using a simple modulus mapping scheme. For example, if the buffers are reused every one hundred packets, the logic to map to the appropriate index would be to divide the sequence number by 100 (modulus 100) and insert the current packet at the index equal to the last two digits of the packet number. In this example the buffer indexes would range from 0 to 99. If the last valid packet seen by the failover device from the primary multicast system is, for example, packet with modulus M, then the failover device will continue broadcasting from the secondary stream buffer with packet with modulus M+1. In this fashion, clients will have continuity in their programs with no discernable interruption.

43. Failover device **16** rewrites the IP headers with a virtual IP address Z.Z.Z.Z and port number ZZ, which is then multicast to clients on the network.

44. The system and method where one program is being multicast in parallel with monitoring by a failover device that has remedial capability has been illustrated.

Those skilled in the art will appreciate that multiple programs may be broadcast at the same time where each program is under the same redundancy and monitoring system using the same equipment as explained above. Thus the present invention should not be limited to the broadcasting of a redundant single stream of data but to the broadcast of multiple streams of data as well. Thus, the present invention is not intended to be limited to one program broadcast at one time. Multiple programs may be run simultaneously.

45. A system and method for multicast video stream failovers has been illustrated. It will be appreciated by those skilled in the art that other variations of the architecture illustrated will be possible without departing from the scope of the invention as disclosed.